

# **Win Forms - Introduction Programming The Rich Client**

**Mark Boulter  
Fred Balsiger  
Microsoft Corporation**

**1-213**

Microsoft®  
**PDC** 2000  
Professional Developers Conference

Microsoft®  
**.net**

the defining

***point***

# Introduction

- **An overview of Win Forms**
  - What is it?
  - What does it give you?
- **Building a simple Win Forms app**
  - Introduction to the Win Forms programming model

# Win Forms

- **What is it?**
  - **A framework for building rich client applications that takes advantage of all the power of .NET Framework and the Win32® API**
  - **An architecture for controls and containers**
  - **RAD design time experience**

# Win Forms

- **“No Touch” deployment**
  - No registration - click, download and run
  - Versionable - no side affects
  - Smart caching and automatic updates!
- **Secure**
  - Can run as semi-trusted browser component
  - Can run as a fully trusted local application
- **Full integration with .NET Framework**

# Demo

- **Win Forms and Web Services**



# Win Forms

- **Rich set of controls**

- **Standard controls:**

Label, LinkLabel, TextBox, RadioButton, Checkbox, GroupBox, Panel, ListBox, CheckedListBox, ComboBox, Button, Menu, Panel PictureBox, MaskedTextBox, Up-Down, Scrollbar, Splitter, Tooltip, Menus, Help, PropertyGrid

- **Common dialogs**

- **RichEdit**

- **COMCTL “Lite”:**

Progress Bar, Status Bar, Toolbar, ListView, TreeView, Tab Control, DateTime Picker, Calendar, Tooltip, Trackbar, ImageList

- **Data controls:**

DataGrid, ErrorProvider

# Win Forms

- **Advanced graphics support - GDI+**
- **“Web look and feel”**
- **Advanced layout**
- **Visual forms inheritance**
- **Printing support**
- **Accessibility**
- **Extensible:**
  - **Easy to add design time behavior**
  - **Extend and enhance the controls and framework**
  - **Drop into direct Win32**



# Demo

- **GDI+ is cool!**

# Win Forms

- **Complete interoperability with ActiveX®**
  - Can host ActiveX Controls
  - Can be hosted as ActiveX Controls
- **International**
  - Runs fully Unicode on Windows NT® and Windows® 2000
  - Runs internally Unicode on Windows 95 and Windows 98
- **Flexible licensing support**

# Security

- **Code access security makes Win Forms controls safe**
- **Code access controlled by:**
  - **Permissions**
  - **Permission Sets**
  - **Code Groups (Zones)**
  - **Code groups have permission sets**

# Browser Integration

- **Controls can be hosted Internet Explorer 5x**
- **Low impact**
  - No installation, registration or GUIDs
- **Secure**
  - Code access security prevents harm
  - No Internet Explorer security dialogs!
- **Lightweight**
  - “Hello World” ~ 3.5K
- **Easily deployed**
  - Dependent components are automatically downloaded

# Demo

- **A Win Forms control in the browser**

# Licensing

- **Extensible licensing model**
- **Complete interop with ActiveX licensing**
- **Design-time and runtime licenses**



# Opening Visual Basic 6.0 Projects In Visual Basic .NET

- Project conversion wizard
  - Copies your Visual Basic 6.0 project
  - Migrates most technologies
  - It's not on the PDC CD
- Some code needs to be rewritten
  - "TODO" comments inserted into code
  - Report generated

# **Building A Win Forms App**

- **Win Forms is about:**

- **Forms**

- **Controls**

- **Events**

# A Simple Form

```
using System;
using System.Windows.Forms;
using System.Drawing;

public class MyForm : Form {

    public static int Main(string[] args) {
        Application.Run(new MyForm());
        return 0;
    }

    public MyForm() {
        this.Text = "Hello World";
        this.BackColor = Colors.SlateBlue;
    }
}
```

# Demo

- **A simple Form**

# A Simple Control

- **To add a Control:**

```
Button button1 = new Button();  
button1.Location = new Point(256, 64);  
button1.Size = new Size(120, 40);  
button1.Text = "Click Me!";  
this.Controls.Add(button1);
```

- **Control state is modeless**

# Events

- Controls raise events

- Handling an Event:

....

```
button1.AddOnClick(new System.EventHandler(button_Click));
```

....

```
private void button_Click(object sender, EventArgs ev) {  
    MessageBox.Show("Hello Win Forms World!");  
}
```



# Event Classes

- **Two Event classes:**
  - **The EventHandler delegate class**
    - Hooks up events to event handling methods
  - **The EventArgs class**
    - Contains information about the event that has been fired
- **Three Event methods on a Control:**
  - **AddOn<EventName>**
  - **RemoveOn<EventName>**
  - **On<EventName>**

# Handling Multiple Events

....

```
Button button1 = new Button() ;
```

```
Button button2 = new Button() ;
```

....

```
button1.AddOnClick(new System.EventHandler(button1_Clicked));
```

```
button2.AddOnClick(new System.EventHandler(button2_Clicked));
```

....

```
private void button1_Click(object sender, EventArgs ev) {  
    MessageBox.Show("Button1 Pushed!");  
}
```

....

```
private void button2_Click(object sender, EventArgs ev) {  
    MessageBox.Show("Button2 Pushed!");  
}
```

....

# Handling Multiple Events

```
....  
Button button1 = new Button() ;  
Button button2 = new Button() ;  
....  
button1.AddOnClick(new System.EventHandler(buttonClicked));  
button2.AddOnClick(new System.EventHandler(buttonClicked));  
....  
  
//The event handling method  
private void buttonClicked(object sender, EventArgs ev) {  
    if (sender==button1) {  
        MessageBox.Show("Button1 Pushed!");  
    } else if (sender==button2) {  
        MessageBox.Show("Button2 Pushed!");  
    }  
}
```

# Demo



# Designer Support

- **Phew! What a lot of code!**
- **But wait – Visual Studio™ does all of this for you....**

## **DEMO!**

# **Building Win Forms Apps**

- **Create a form**
- **Add your controls**
- **Handle events and add your code**
- **Great designer support!**



# Visual Inheritance

- Win Forms is based on inheritance
- Visual Studio supports designing inherited forms

# DEMO!

# Summary

- **Win Forms is the way to write rich client controls and applications**
- **Win Forms controls are secure, lightweight, and support hosting on web pages**
- **Its easy to write really great Win Forms apps!**

# **Related Sessions And References**

- **Other PDC sessions recommended:**
  - **Win Forms Advanced (1-402)**
  - **Security Considerations for Download Controls (1-222)**
  - **GDI+ and Win Forms (1-212)**
  - **Deploying and Versioning Your Application (3-200)**
  - **Web Services Platform Security (3-300)**
- **Win Forms Quick Start in the SDK**

# Questions?

The background is a solid green color with a subtle gradient. In the lower-left quadrant, there are three faint, light-green circles of varying sizes, connected by thin, curved lines, creating a minimalist diagram or network structure.

Where do **you** want to go today?

**Microsoft**